

## Cromemco Software Update Service Note Cromix-2

Date: September 22, 1981

Product: CROMIX-L and CROMIX-S

Version.release: 11.03 and 11.05

Date production of this version began: September 22, 1981 on 8"  
September 22, 1981 on 5"

First serial number with this version: CX1611698 on 8"  
CX1710938 on 5"

[Note: Earlier serial numbers may also contain this version. All Cromemco Finished Goods stock was recopied on the above dates.]

### Description of enhancements, corrections, and known problems:

Cromix version 11 is now available, along with a revised Cromix Instruction Manual, part number 023-4022, August, 1981. This release of the Cromix Operating System has many new Shell commands, utility programs, and system calls, and the following is only a brief description of those modifications and new features. Most of the topics mentioned below are discussed fully in the new release of the Cromix Instruction Manual, which should be used for further reference if needed.

This note contains an index which is comprehensive over all Cromix SUDS notes. Page numbering starts where it left off on the preceding note.

**PLEASE DO NOT BEGIN USING THE ENCLOSED  
CROMIX DISK UNTIL YOU HAVE READ  
THE NEXT 3 PAGES.**

### Important Points About Cromix Version 11

This release of the Cromemco Cromix Operating System contains many new and powerful features, including an on-line manual, pipelines and signals, prioritized execution of processes, record-level lock, Quadart/IOP and accurate clock driver support, and multiple processes per bank of memory. However, there are also several important points to be aware of to ensure that you achieve satisfaction with this product. These have been drawn from the documentation and collected here for easy reference.

1. This version of the Cromix Operating System is supplied on double-sided disks only. This extra amount of disk space is required to store all the various programs and help files now supplied with the Cromix Operating System. The 8 inch disks have very little free space left on them. Therefore, it is recommended that you immediately make a backup of the enclosed disk in one of the following ways:
  - a. transfer all programs and files to a hard disk by means of the Update or Newdisk facility,
  - b. create a double-sided, double-density floppy disk by means of the Newdisk facility, or
  - c. create two double-sided floppy disks, one of which will be the root disk and contain all programs and the other of which will contain all of the help files.

The last procedure will have to be done manually. Five inch Cromix disks are already supplied in this last format, one disk containing the programs and the other containing the help files. To use them together, type **mounthelp** once you have booted up the Cromix system with the main disk. This will mount the drive b disk to the directory /usr/help of the root disk. You also may elect not to use the on-line manual facility and thus work with only the main Cromix disk.

Cromemco Software Update Service Note  
Cromix version 11.03

2. Due to the new, more accurate clock mechanism being used in Cromix version 11, the main console terminal (the one used to boot the disk) **can not** be set to a baud rate of 19200. The main console may be set to a baud rate of 9600 or lower, and any additional consoles connected to the system may be set to a baud rate of 19200 or lower.
3. A Crogen (Cromix generator) facility is now supplied with Cromix version 11. This facility and its support files are stored in the /gen directory. The Cromix system with which the enclosed disk will boot up has been generated including TUART console drivers and Cromemco standard printer drivers, but not including QUADART drivers or serial printer drivers. Crogen allows you to select a default root device, preventing you from having to specify the root device each time you boot up.
4. Cromix version 11 is capable of being booted up and run directly with IOPs and QUADARTs. The enclosed disk may be converted for this operation by means of the **Rungd** facility (see on-line manual). You may also select this operation by creating a new Cromix System with Crogen. However, the 16FDC board must be configured for automatic startup operation since no terminal will be connected to it. This is done by setting switches 3 and 5 of the 16FDC to ON. Also, you **must** configure the Cromix system with a default root device if booting up with an IOP/QUADART and if no terminal is connected to the 16FDC (use default program or Crogen).

There are several differences between Cromix version 11 and previous versions which require that you change any old Cromix disks you wish to continue using (including hard disks).

5. The password encryption scheme used in Cromix version 11 is greatly improved and more secure than that used in previous versions of the Cromix System. This means that the /etc/passwd file of any old Cromix disks **MUST HAVE ALL PASSWORDS REMOVED FROM IT** while

running the previous version of the Cromix Operating System. These may then be replaced while running the enclosed version of the Cromix system. The new passwords use an encryption key; therefore, do not be concerned if the same password produces two different encryptions.

6. The structure of device drivers is slightly different in version 11. Each driver now has associated with it both a major and a minor device number. The minor number for character devices often corresponds to the port address of that device. (For example, lpt1 has the minor device number 5, meaning that that printer is connected to base port address 50H.) You will also be required to give both the major and minor device numbers when booting up the enclosed disk.
7. The user login procedure has been separated out from the main portion of the Cromix Operating System to make more room in the system bank of memory. It is now a separate program (login.bin) stored in the /etc directory. **THIS PROGRAM MUST BE IN THE /ETC DIRECTORY** on all disks from which you would like to boot up Cromix version 11.

Please refer to the Cromix manual for additional information about Cromix version 11. It is also recommended that you read the new user information on the enclosed disk before attempting to use it any further. The Update facility provided is the recommended way for updating your old Cromix disks.

## Enhancements to the Operating System

### New Shell Commands

#### Pipes and sequential pipes

Pipes and sequential pipes have been implemented in version 11 of the Cromix Operating System. The pipe symbol is the vertical bar ( | ). The sequential pipe symbol is the less than, greater than sign ( >< ). Pipes and sequential pipes send the standard output of one process to the standard input of another process. Sequential pipes achieve the same effect by sending the output of the first command to a temporary file and then from the temporary file to the input of the second command. See the section on redirected I/O in the manual for further details.

#### Redirection of error messages

The output that a process sends to standard error can now also be redirected, in addition to the standard output, using the greater than sign immediately followed by a star ( >\* ). The star can also be used immediately after the pipe and sequential pipeline symbols ( |\*, ><\* ). See the Cromix Instruction Manual section on redirected I/O for further information.

#### tee

A facility known as a tee can be used with either the standard pipe or sequential pipe to send the output of a pipe to the standard output and to a file simultaneously. For example:

```
% list | tee file1
```

lists the current directory and sends that listing to the file **file1** as well as to the console.

#### if

A new Shell command which allows the programmer to write command files that execute commands conditionally is the If command. For example:

```
command1  
IF -err command2
```

In this example **command2** is executed if **command1** returns an error code when it terminates.

### **kill**

Another new Shell command is the Kill command which sends a specified signal to a specified process. If the signal type is not specified, Kill will default to a terminate signal. Kill -2 1, for instance, will kill all processes and shut the system down. Kill 0 will abort all background jobs attached to a user's terminal.

### **path**

The Shell command Path now exists to provide a convenient way to find the directory location of an executable file or command file. If the command specified is a Shell command, Path will notify the user of that fact also.

### **sleep**

The Sleep Shell command will suspend execution for the number of seconds specified as an argument.

### **Control-C**

Entering Control-C from the keyboard will send an abort signal to the current process, which terminates execution of the process unless the process has used the system call `.signal` to disable or catch the abort signal.

## New System Calls

### **.alarm**

The system call **.alarm** sends an alarm signal to the current process after the specified number of seconds has elapsed.

### **.exchg**

The filenames of two open files can now be exchanged with the system call **.exchg**. After **.exchg** has been executed, the two filenames remain associated with their original inodes but the block pointers of the inodes will be changed. Chapter 8 of the Cromix Instruction Manual contains a complete description of the inode structure which may be useful in understanding how **.exchg** functions.

### **.getprior**

The **.getprior** system call returns the priority of the current process. This number is in the range -40 (highest priority) to 40.

### **.kill**

The system call **.kill** sends a signal to a process. When any signal is received by a process, the process is aborted unless the signal system call has been used to specify that a subroutine be executed or that the signal be ignored.

### **.lock**

The **.lock** system call provides a facility to lock a record or file. This system call does not explicitly provide a locking mechanism but aids the programmer in implementing a system that achieves mutual exclusion on files.

### **.pause**

The **.pause** system call is another interprocess communication related facility. **.Pause** will suspend execution of the current process until a signal is received from kill or alarm.

### **.pipe**

The **.pipe** system call will create and open an input and output channel which may be passed to a child process for interprocess communication.

### **.setprior**

Process priorities can be changed using the system call **.setprior**. **.Setprior** will change the current process priority as specified with the priority number in the range of -40 (the highest priority) to 40. Only a privileged process may set a priority in the range of -1 to -40.

### **.signal**

The system call **.signal** sets up a process to receive a signal. When a specified signal is received, **.signal** will pass control to the address specified when the **.signal** system call was made. The program will return to the point of execution where the signal was made when it encounters a RET instruction.

### **.unlock**

The **.unlock** system call has been provided to unlock a file locked by the **.lock** system call.

## **Modifications to the Utilities**

### **blink 00.12**

The Blink program is a two pass virtual linker. Blink manages memory in a way that allows it to link programs up to the total amount of memory available. The space that the linker occupies in memory during execution does not impose a restriction on the size of the program being linked. Thus, Cromix programs up to 64K minus the 1K of memory that the Cromix Operating System occupies in each user bank can be linked by Blink. CDOS programs running under the Cromix Operating System are limited to approximately 4K less memory than the 63K available to Cromix programs because of the size of **sim**, the CDOS simulator, which must also be loaded. It should be noted that Cobol programs using segmentation cannot be linked using Blink.



### **crogen**

Cromix version 11 contains a facility for generating a **cromix.sys** file tailored for an individual user's machine configuration. The program for doing this is called Crogen and is located in the /gen directory along with the libraries needed.

### **default**

The Default program will allow the Cromix Operating System to use a default root device and login name when booting the operating system.

### **compare -t 00.07**

The **-t** option has been added to Compare to allow comparing files in terse mode. In terse mode, Compare informs the user whether or not the two files are identical, but does not print out the differences.

### **day 01.01**

The Day program executes the specified command if it is currently the specified day.

### **deltree 00.03**

The Deltree program deletes all filenames and directories in a tree including the specified directory and also all child directories and files. Normally, Deltree will prompt the user with the file name or directory name and (y,n). If the user types **y**, the file or directory will be deleted; if **n** is typed, it will not be deleted.

### **echo 00.04**

Two new options can be used with the Echo utility. The Echo program echoes its arguments to the standard output. The **-e** option echoes the string specified to standard error instead of standard output. The **-n** option will echo a string without terminating it with a newline character.

#### **find 00.06**

The Find program searches the specified directory according to an expression. For example:

```
% find / -name "*.bak" -a -exec del -v {}
```

finds all files named anything.bak and deletes them.

#### **help 00.02**

The old Cromix Operating System Shell command Help is now a greatly expanded utility program which accesses a large on-line manual. The Help program allows the user to select from a menu of all the available help topics when called without an argument. When a given topic is selected, the Help program will allow the user to view the file with the ability to move forward and backward in the file. At present only the manual entries for the Shell commands and utility programs are available on line. The data base for the on-line manual is in the directory /usr/help. The file /usr/help/msg.msg contains the messages printed on the bottom line of the screen when the Help program sends a file to the console. The msg.msg file is linked to the file msg2.msg which contains messages that use the attributes of the Cromemco 3102 terminal. If your system uses the Cromemco 3101 terminal, then the file msg1.msg should be linked to the file msg.msg. This can be achieved by typing the following command:

```
maklink -f /usr/help/msg1.msg /usr/help/msg.msg
```

#### **input 01.00**

The Input utility reads a string from the standard input and upon reading a newline, sends that string to the standard output. This utility can be used to write interactive command language programs by redirecting the output of the utility to a file and then testing the contents of the file with the utility Testinp. Refer to the on-line manual entry concerning command files using Input and Testinp. Input reads a maximum of 255 characters from the standard input and 512 characters if input is redirected from a file, and terminates reading upon reading a newline.

Example:

```
% input > temp
```

This command line will read one line from the standard input and send it to the file **temp**.

**l -s 00.11**

The L utility program, used for listing a directory's contents, has a new option. The option is **-s** and it generates a summary of disk space used.

**match 00.03**

The Match utility will search through the files specified in the file list for all occurrences of the string specified and display each line containing a match.

**mode 01.01**

The Mode utility program displays or alters the operational characteristics of a character device. This version of the Cromix Operating System has several new modes; refer to the manual for information on the new modes. If Mode appears in the command line without any arguments, the utility will display the current operational characteristics of the device from which the system received the mode call.

**root 00.02**

The Root utility program displays the device which contains the root directory.

**sort 00.06**

A useful and flexible sorting program for ASCII files is now a utility. Sort sorts or merges lines of one or more files together. There are several options that can be used with Sort to control the sorting order or select the key on which to base the sorting.

**spool -q -h -m 00.09**

The Spool utility program now has three new options. The **-h** option prints a header for the specified file when it is being printed on the

specified device. The **-m** option followed by a number will send a file that number of times to the specified device. The **-q** option will kill the spooling of all files attached to the specified device.

### **testinp 01.01**

The Testinp utility compares the contents of a file with a string or strings and sets an error return code if the strings do not match the contents of the specified files. The test Testinp makes is case insensitive. A test string in upper, lower, or mixed case will match a string in upper, lower, or mixed case. The **-r** option reverses the sense of Testinp by setting the error code if a match does occur. The **-f** option will check only the first character of the file passed as an argument against the first characters of the control strings. The **-d** option will delete the file passed as an argument after the test is made. This option is useful in many command files that use a temporary file created during the execution of the command file. Refer to the on-line manual entry on the Input utility regarding its use in command files.

Example:

```
echo -n "Do you want to shut down the system?"
input > temp
testinp temp YES OUI SI
if -err goto noshutdown
kill -2 1
%noshutdown
del temp
```

The above example is a typical command file that shows the use of Testinp and Input. The first line sends the string within the quotes to the standard output. The second line uses the Input utility to send the user's response to the file **temp**. On the third line Testinp is used to test the contents of the file **temp** for the occurrence of the strings **YES**, **OUI**, or **SI**. If the file contains one of these strings, then the system will be shut down using the Kill command. If the file **temp** does not contain an occurrence of any of the strings mentioned above, then Testinp will set an error code and the following If command would pass control to the label **noshutdown**, skipping the Kill command. In other words, if the user of the program answered no to the question generated, then the system would not be shut down. The last line

deletes the file **temp** created by the Input utility on the second line of the command file. If the **-d** option had been used with Testinp, then the file would have been deleted automatically by the Testinp program.

#### **unmount 00.10**

The Unmount command can now be used with a **-x** option that will unmount the large floppy disk without ejecting it from the drive.

### **New Command Files**

#### **check and check -s**

The Check command runs the programs Dcheck and Icheck on the file system. Check should be run after rebooting the system after a system crash, or any time that the integrity of the file system is in doubt. The Startup command file program which is executed after every boot up is designed to indicate when the Check program should be run. Refer to the write up on the command file Startup in this section for more information on this function. The **-s** option to Check does the following: **dcheck -s; ickcheck -s; boot.** The **-s** option is the salvage option used with Dcheck and Icheck to repair most file system problems. Refer to the Cromix Instruction Manual for more information on Dcheck and Icheck.

#### **mounthelp**

The Mounthelp command will mount the second Cromix Operating System 5 inch diskette into the directory **/usr/help**. The second disk contains the files used by the Help program only. The Cromix Operating System can be used without mounting the second disk with no difficulties except for the inability to access the on-line manual which resides on disk two.

#### **newdisk**

The program Newdisk now exists for making copies of the system disk. Newdisk must be followed by the name of the device on which the new disk is to be created. The Newdisk command file will first execute the Init program. Be sure to specify the correct disk drive as all of the data on the specified disk will be destroyed.

### **newuser**

The Newuser program displays a file with information useful to new users.

### **query**

The Query utility program provides a quick way to find the names of commands. Query uses key word lookup using a file containing one line descriptions of all the commands as a data base. If the -s option is used, Query will find the appropriate system calls in addition to Shell commands and utility programs. If additional information is needed on the command, the user can access the Cromix Instruction Manual or its on-line version by typing **help** followed by the command name.

### **runtu and runqd**

This version of the Cromix Operating System is equipped with terminal drivers for the IOP and Quadart. The Runqd command file contains the necessary commands to configure the Cromix Operating System to use the IOP and Quadart. The Runtu command file contains the necessary steps to return the Cromix Operating System to the original 16FDC or TUART configuration. Additional information concerning these programs can be found by running the Newuser program.

### **startup**

The file **startup.cmd** remains basically unchanged since previous versions except for one modification. After the system is booted, the Startup program determines if the system was previously shutdown in the correct manner with the use of the Shutdown program. If the system was not previously shut down in that manner, the Startup program will inform the user that the Check program should be run to verify the integrity of the file system. For example, if the system crashed due to a power failure or runaway program, then upon rebooting the system the Startup program will send a message to the console recommending that the Check program be run. Refer to the documentation on the command file Check for further information on this facility.

### **shutdown**

The Shutdown program contains the commands to shut down the operating system by killing all processes, flushing all buffers, and logging off all users. The Shutdown program contains a facility that works with the Startup program to detect inadvertent system terminations. To make use of this facility it is recommended that the Shutdown program be run whenever the operation of the system is to be terminated.

### **update**

The Update program updates a Cromix system disk with a newer Cromix system disk.

### **install**

A command file now exists that simplifies the copying of CDOS disks over to a Cromix disk. Typing **install** followed by the device name that contains the CDOS disk will transfer the necessary CDOS programs to the appropriate directories on the Cromix disks. Cromemco CDOS disks are gradually being equipped with this facility.

### **The New Manual**

The new **Cromix Instruction Manual** has been expanded and modified considerably since the previous manual. The Getting Started section for new users has been improved. A new section explaining the disk allocation system has also been added. Many other sections have been rewritten to cover the new and expanded features of the Cromix Operating System.

### **Updating to version 11**

If you are currently running Cromix version 10 and wish to update to version 11, then pay close attention to these instructions. There are important differences between Cromix versions 10 and 11. The easiest way to update your present Cromix system is to boot up this new disk and then type the command **update** followed by a block device name to update your old disk. This method will not preserve many of your old Cromix files. This process could be done manually with the Copy and Cptree programs, but this is not recommended.

The major differences between the two versions of the operating system occur between the contents of the new

and the old `/bin` and `/dev` directories. Devices now have associated with them both a major and minor device number; hence, Cromix version 11 requires a completely new `/dev` directory. You can preserve your old `/dev` directory by renaming it `olddev` and then making a new `/dev` directory.

Many of the Cromix version 11 bin (executable) programs will not execute correctly and differ from their equivalents under Cromix version 10. Again, you can preserve your old bin directory if you wish by renaming it `oldbin` and copying the Cromix version 11 bin programs to a new `/bin`.

### **Password encryption**

A vastly more secure password encryption scheme has been implemented in Cromix version 11 to improve system security. Since the password encryption is different, a user WILL NOT be able to log in on a disk under the new Cromix Operating System unless the `/etc/passwd` file contains at least one privileged user name without a password. This is very important so that one can initially log in on an old disk with Cromix version 11.

The recommended procedure is for a privileged user of the system first to clear the password file of all users, bring up the new Cromix Operating System, and then restore all users' passwords by means of the `Passwd` program. Also note that the new owner of most of the Cromix system software is `bin`, owner number 32767, in the `passwd` file. This user name and ID should be added to your `passwd` file.

### **System clock**

If your system has a 16FDC disk controller, then you can take advantage of a much more accurate clock in Cromix version 11 than prior versions. The file `/etc/ttys` must be modified and a jumper connected between the two RTC pads on the 16FDC. Refer to the version 11 Cromix manual and Chapter 3 of the 16FDC manual for more information.

### **Multiple processes per bank**

Multiple processes can now be run in one bank of memory, provided those processes are created to do so. Users can create relocatable programs using the `-r` option to `Blink`. Refer to the manual entry on `Blink` for more information. Many of the bin utilities on this disk have been created with a bitmap to allow this



relocation. However, not all programs are eligible for this treatment. In particular, programs such as compilers, assemblers, and editors require a bank to themselves because of the amount of memory they use. Also, CDOS programs running under the Simulator will use the remainder of a bank.

### **Argument list size**

Under the Cromix Operating System, programs can be passed an argument list no longer than 512 bytes. Occasionally a program will be passed an argument list longer than what is permitted, resulting in the generation of the error "Argument list too big". In most cases the program will continue to execute with a truncated argument list. Often the effect of a particular program will be modified by an argument list that is smaller than what was intended. It is advisable to kill any program that generates such an error with control-c, and then determine if a truncated argument list will cause any undesirable effects.

### **Modifications made to jsysequ.z80**

Several modifications and additions have been made to the `jsysequ.z80` and `modeequ.z80` files. The modifications made to the `jsysequ.z80` file are the following:

The C register file position modes for `.setpos` have new names. The new names are: `fwd.begin`, `fwd.current`, `fwd.end`, `bak.current`, and `bak.end`.

`St.devno` is a new equate for use with `.fstat`, `.cstat`, `.fchstat`, and `.cchstat`. `St.devno` is used when reading or changing device numbers.

The following equates have been added for use with signal related system calls: `sigabort`, `siguser`, `sigkill`, `sigterm`, `sigalarm`, `sigpipe`, `sighangup`, `.signal`, `.kill`, `.sleep`, `.alarm`, and `.pause`.

Other additions to the `jsysequ.z80` file include: `.getprior`, `.setprior`, `.getproc`, `.ksam`, `.lock`, `.unlock`, `.fork`, and `.memory`.

The error code definitions use `DEFV` rather than `DEFS` to reserve storage space.

### Modifications made to modeequ.z80

The following new equates have been added:  
**Md\_fnkeys**, **md\_iflush**, **md\_moded**, and **md\_ispeed**.

**Md\_ispeed** has the following options available to it: **s\_hangup**, which hangs up the phone when the device is finally closed; **c\_ctswait**, which allows a wait for Clear To Send; **s\_uninit**, which indicates that the baud rate has not been initialized yet; and **sfl\_auto**, which will cause the baud rate to be set automatically after carriage returns are sent from the keyboard.

**Md\_moded** is the number, in tenths of a second, of delay inserted after the characters CR, LF, TAB, and BS in the Quadart drivers **qtty** and **qslpt**. Furthermore, in the **qslpt** driver, eight times the delay is inserted after a formfeed. The previous names **nlDelay**, **tabdelay**, **crdelay**, **ffdelay**, and **bsdelay** have been changed to **nl**, **tab**, **cr**, **ff**, and **bs** respectively.

**Md\_model** now has equates for the options **odd** and **even** for setting, checking, or stripping parity bits.

**Md\_mode3** now has the option **discard**. If that bit is set, the space allocated for that device will be discarded, making room for other devices after the device is closed.

### Writing Command Files

Any command or sequence of commands that can be entered on the command line can also be put in a file and executed by entering the name of that file on the command line. A file which contains one or more commands to the operating system is known as a **command file**. A command file name must have the extension **.cmd** and must reside in the current directory, or the directory **/cmd** to be automatically found by the Cromix Operating System for execution. Once the command file is written it can be used like any other command by simply entering the name of the file, less the **.cmd** suffix, on the command line. The ability to add user defined commands to the operating system gives the user the power to customize the Cromix Operating System to virtually any application.

Parameters can be passed to any command file from the command line by referring to those parameters in the command file by a **#** followed by a number. The number

refers to the specific parameter on the command line. Read the section on argument substitution in the Cromix Instruction Manual for more details on this feature of the Shell. In a command file, jumps and conditional jumps can be made to labels by using the Goto and If Shell commands. Command files can send output to the standard output using the Echo program. Refer to the Cromix Instruction Manual for information on the use of these programs and Shell commands. The following example is a listing of the command file **echo\_args.cmd** and illustrates the use of some of these Shell commands:

```
%start
if .#1 = . go done
echo #1
shift
go start
```

Let us assume that the following was typed on the command line to invoke execution of the command file **echo\_args.cmd**.

```
% echo_args Hello world
```

This command will invoke the execution of the command file **echo\_args.cmd** with the argument **#1** being **Hello** and argument **#2** being **world**. The first line of the command file is **%start** and acts only as a label producing no effect. The second line uses the If Shell command to test whether or not the string produced by concatenating the argument **#1** to the end of the character **.** equals the string which is just the **.** (period). At this point, the string **.#1** expands to **.Hello** and does not equal the string **.**, so the condition is false and control passes to the next line. If no arguments had been given on the command line that called **echo\_args**, then argument **.#1** would expand to just **.** (the period by itself), and the test **if . = . go done** would be true and control would pass to the label **done**. In this file there is no label **done**, so control would jump to the end of file, which is the default when a label is used but not declared. The test in the present case, however, is false and the next line is executed. This line expands into **echo Hello** which causes the string **Hello** to be sent to the console on a line by itself. The next line will now be executed, which is the Shift command. Shift moves all the arguments in the argument list to the left one place. Argument **#1** will now be the string **world**. The following statement is a Goto to the label **start** which will repeat the sequence described above except that the string **world** will be printed by Echo this time through the loop. The argument list will again be shifted by one and control will again jump to the

Cromemco Software Update Service Note  
Cromix version 11.03

beginning, but this time the If command will transfer control to the end of the file which will terminate execution of the command file and return control to the operating system.

ERRATA/ADDENDUM

Manual : Cromemco Cromix Instruction Manual  
Publication Date : August 1981  
Part No. : 023-4022

----

Please make the following manual corrections:

Page : 62, between init and kill -1 1  
Append : ioprun loads a program into an IOP

Page : 121, fourth line  
Incorrect : iopru!  
Correct : ioprun

Page : 135, end of first paragraph  
Incorrect : a space, a backspace, and a space.  
Correct : a backspace, a space, and a backspace.

Page : 135, section heading ECHO  
Incorrect : ECHO  
Correct : ECho

Cromemco Software Update Service Note  
Cromix version 11.03

Pages : 136, 323, parity bit table  
Correct :

EVEN	- disabled}	Strips parity bit and does not check for parity. Space parity, bit 7 always equals 0.
ODD	- disabled}	
EVEN	- enabled }	Strips parity bit and requires even parity. Also transmits even parity.
ODD	- disabled}	
EVEN	- disabled}	Strips parity bit and requires odd parity. Also transmits odd parity.
ODD	- enabled }	
EVEN	- enabled }	Does not strip parity and does not check for parity. Receives and transmits all eight bits.
ODD	- enabled }	

The Cromix Operating System does not allow parity bit to be = 1. If the terminal internally makes bit 7 = 1, the operating system will not accept the text.

Page : 308, last line of example  
Incorrect : '/nThe number = %d/n',0  
Correct : '\n\nThe number = %d\n',0

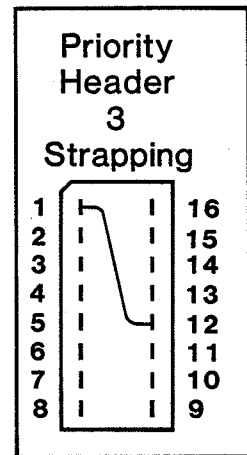
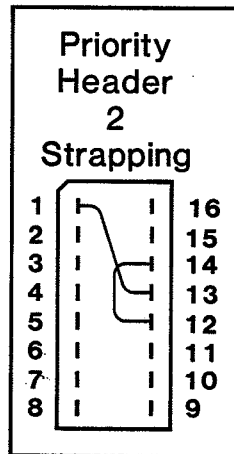
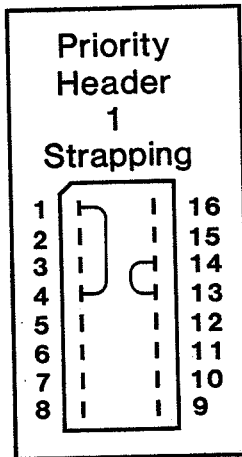
Page : 350  
Incorrect : stdin  
Correct : stdout

Page : 359, third paragraph  
Incorrect : Chapter 6  
Correct : Chapter 3

Cromemco Software Update Service Note  
Cromix version 11.03

Pages : 369, 371, 373, and 375, the Priority Header graphics

Correct : Note that the header should be inserted into the socket provided for IC28. No header is required on Quadart 4.



Page : 406, under the K listing

Incorrect : kill system call, 278  
kill utility, 377

Correct : kill command, 122  
kill system call, 278

Cromemco Software Update Service Note  
Index

-err, 13

.alarm, 15  
.exchg, 15  
.getprior, 15  
.kill, 15  
.lock, 15  
.pause, 15  
.pipe, 16  
.setprior, 16  
.signal, 16  
.unlock, 16

Argument list size, 25

blink, 16

compare -t, 17  
Control-C, 14  
crogen, 17  
Cromix Instruction Manual, 23

day, 17  
default, 17  
deltree, 17

echo, 17

find, 18

help, 18

if, 13  
input, 18  
install, 23

kill, 14  
Kill -2.1, 14

l -s, 19



Cromemco Software Update Service Note  
Index

match, 19  
mode, 19  
Multiple processes, 24

newdisk, 21  
newuser, 22

Password encryption, 24  
path, 14  
Pipes and sequential pipes, 13

query, 22

Redirection of error messages, 13  
root, 19  
runtu and runqd, 22

shutdown, 23  
sleep, 14  
sort, 19  
spool -q -h -m, 19  
System clock, 24

tee, 13  
testinp, 20

unmount, 21  
update, 23  
Updating to version 11, 23